

Minikube

In the context of DevOps, Minikube is a lightweight, single-node Kubernetes cluster tool used for local development and testing, allowing developers to experiment with Kubernetes features without the complexities of a full-fledged cluster.

=> MINIKUBE

```
--> curl -LO
```

```
https://dl.k8s.io/release/v1.32.0/bin/linux/amd64/kubectl sudo
```

```
install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

```
--< --> chmod +x kubectl mkdir
```

```
-p ~/.local/bin mv ./kubectl
```

```
~/.local/bin/kubectl
```

```
--< kubectl version --
```

```
client
```

```
-->
```

```
curl -LO
```

```
https://github.com/kubernetes/minikube/releases/latest/download/minikubelinux-  
amd64
```

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-  
linuxamd64
```

```
--< minikube
```

```
start
```

docker ps (check is container is running) kubectl

get node

kubectl get pod (No resources found) kubectl

get pod -o wide (No resources found)

minikube status kubectl get deploy kubectl

get replica

JENKINS pipeline

pipeline {

agent any

tools {

maven 'Maven'

}

stages {

stage('SCM') {

steps {

git branch: 'main', url:

'https://github.com/NavinkumarD/DevOps-

1.git'

bat 'dir' // Check directory structure

after clone

```
}  
}
```

```
stage('Build') {  
  steps {  
    dir('simple-web-app') {  
      bat 'mvn clean install'  
    }  
  }  
}
```

```
stage('Build to Images') {  
  steps {  
    dir('simple-web-app') { // Ensure we  
are in the correct directory  
      bat 'docker build -t  
22csl258/mysimplewebapplication.'  
    }  
  }  
}
```

```
stage('Push to Hub') {  
  steps {  
    script {
```

```
docker.withRegistry('https://index.docker.io/v1
```

```
/', 'docker-hub') {
```

```
docker.image('22csl258/mysimplewebapplicat
```

```
ion').push('latest')
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
havin@ITP-CC16-42: /mnt/c/Users/Admin$
get:10 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
get:12 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
get:15 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.1 kB]
get:16 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [8076 B]
get:17 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7076 B]
get:18 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [272 B]
get:19 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [26.4 kB]
get:20 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [16.3 kB]
get:21 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
get:23 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [21.3 kB]
get:24 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1304 B]
get:25 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
get:26 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 1998 kB in 2s (837 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
havin@ITP-CC16-42:/mnt/c/Users/Admin$ minikube start
* minikube v1.35.0 on Ubuntu 24.04 (amd64)
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Restarting existing docker container for "minikube" ...
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
* Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
  * Enabled addons: default-storageclass, storage-provisioner
  * Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
havin@ITP-CC16-42:/mnt/c/Users/Admin$ kubect1 get node
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready     control-plane   22h   v1.32.0
havin@ITP-CC16-42:/mnt/c/Users/Admin$ minikube start
* minikube v1.35.0 on Ubuntu 24.04 (amd64)
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Updating the running docker "minikube" container ...
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
* Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
  * Enabled addons: default-storageclass, storage-provisioner
  * Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
havin@ITP-CC16-42:/mnt/c/Users/Admin$ minikube start
(check is container is running)
kubect1 get node
kubect1 get pod (No resources found)
kubect1 get pod -o wide (No resources found)
```